ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

Título del proyecto:

SISTEMA DE SUPERVISIÓN DEL USO DE LA RED PARA PYMES

Oscar Echeverría Esparza
Eduardo Magaña Lizarrondo
Pamplona, 14 Septiembre de 2010

ÍNDICE

RESUMEN

Hoy en día, las pequeñas empresas se sirven de los medios tecnológicos para desarrollar su actividad, y algunas de ellas no solo se sirven de dichos medios tecnológicos sino que dependen de ellos. Entre esta tecnología,, se encuentran las herramientas informáticas y de comunicación de las cuales hacen uso sus empleados, entre las que se destaca el correo electrónico y la conexión a Internet.
Sin embargo, estas herramientas informáticas puestas a disposición de los empleados para desarrollar su trabajo, al tiempo que favorecen la actividad laboral, también pueden convertirse en un perjuicio para la empresa.
Debido a esto anterior se desea conocer el uso que los empleados de una pequeña empresa dan a sus máquinas.
Existen dos posibilidades para llevarla a cabo: Una sería probar plataformas existentes open-source , como http://www.ntop.org para familiarizarse con este tipo de aplicaciones. También existen si se quiere otras plataformas de pago. Se ha elegido como herramienta a probar el Ntop (un analizador de tráfico de red). La prueba de Ntop consiste en ver cuántos paquetes por segundo es capaz de procesar, es decir, a partir de qué número de paquetes por segundo comienza a tener pérdidas. Para ello se ha creado un cliente y un servidor con Iperf (permite crear flujos de datos UDP y TCP) para inyectar tráfico a la red y así poder poner a prueba a Ntop capturando dichos paquetes.
La otra posibilidad se trata de desarrollar un sistema propio que a través de una Web ofrezca información de qué puertos son los más usados en la red, perfil de actividad de cada usuario mostrando el tráfico entrante y saliente, si usan aplicaciones P2P, etc. Esta aplicación se ha dividido en 4 módulos:

- Módulo de captura: obtiene los paquetes de la tarjeta de red, y ha sido realizado mediante un programa en el lenguaje C. Ha sido necesario utilizar la librería libpcap para poder comunicarse con la tarjeta de red.
- Módulo de procesado: obtiene de cada paquete los datos que nos interesan, tales como IP origen, puerto origen, tiempo de llegada etc. También ha sido realizado en lenguaje C.
- Almacenamiento: guarda en una BD la información que nos interesa de cada paquete. También se ha realizado en el lenguaje C pero utilizando la librería MySQL para C para poder conectarse con la BD. Antes de decidir guardar la información en una BD de datos se han realizado unas pruebas para ver si era suficiente en vez de utilizar ficheros.
- Interfaz Web: muestra en una página Web la información de la BD mediante tablas y gráficas. Se ha implementado en PHP junto con MySQL (para conectarse con la BD). También es necesario el Apache que actúa como servidor Web.

Al finalizar el prototipo, éste ha sido evaluado con las mismas pruebas que se hicieron sobre Ntop, y así poder compararlos y ver sus diferencias.

1.INTRODUCCIÓN

Internet ha tenido un fuerte desarrollo en los últimos años y ha influido mucho en la vida de los trabajadores, en el ocio, y en el conocimiento a nivel mundial. Gracias al desarrollo de la tecnología muchísimas personas tienen acceso a él y por tanto a toda la información de circula a través de la red.
Internet ha tenido un profundo impacto sobre las personas. Gracias a la Web, millones de personas tienen acceso fácil e inmediato a una cantidad extensa y diversa de información en línea.

Con la aparición de Internet y de las conexiones de alta velocidad disponibles al público, Internet ha alterado de manera significativa la manera de trabajar de algunas personas al poder hacerlo tanto desde el trabajo como desde sus respectivos hogares. Ha permitido a estas personas mayor flexibilidad en términos de horarios y de localización, contrariamente a la jornada laboral tradicional de 9 a 5 en la cual los empleados se desplazan al lugar de trabajo. Pero sin duda alguna, hoy nos podemos perder por el inmenso abanico de posibilidades que nos ofrece la Red. Por eso es necesario poder disponer de herramientas que permitan detectar el buen o mal uso de éste.

Hasta hace poco los usuarios han podido funcionar bien con dispositivos Ethernet 10/100, pero ahora es necesario monitorizar redes de mayor capacidad debido al avance de las tecnologías (la aparición del ADSL), y nuevas aplicaciones que requieren más ancho de banda para su correcto funcionamiento, lo que ha provocado que se desarrollen las tecnologías de Gigabit Ethernet y 10 Gibabit Ethernet.

Este proyecto se basa en familiarizarse con herramientas que permitan supervisar el uso que los usuarios de una empresa hacen de Internet a alta velocidad y también la creación de una aplicación similar a éstas pero que sea fácil de manejar por todo el mundo, incluso por gente no muy familiarizada con la informática.

El entorno donde se va a desarrollar la aplicación es el de una pequeña empresa, en la cual sus trabajadores van usar de manera frecuente Internet para la correcta realización de su trabajo.

Pero el tema de obtener datos que circulan por tu red es muy espinoso en cuanto a la intrusión de la intimidad de las personas.
Como norma general, no se tiene derecho a mirar el contenido del tráfico de los usuarios de tu red, aunque fuera un familiar y aunque no vaya cifrado. Sin embargo, teniendo en cuenta que usan tus recursos y que por tanto podrían cometer delitos en tu nombre se debería poder conocer todo ésto. Se trata de un asunto muy discutido y muy complicado de tratar, del que hay diversidad de opiniones. No se va a entrar en este tema.

1.1 OBJETIVO

El objetivo final de este proyecto será la elaboración de una herramienta que permita visualizar estadísticas sobre el uso de las maquinas en sí que llevan a cabo los miembros de una misma red. Es decir, la interfaz gráfica mostrará el uso que estos usuarios realizan en sus propias máquinas, dará información acerca de qué aplicaciones usan (messenger, emule, Web..). Para ello previamente se deberá de elaborar alguna aplicación que permita obtener datos, a partir de los paquetes que pasen por la tarjeta de red [10] de la máquina en la que se instale la aplicación, para la posterior elaboración de las estadísticas que nos informen del uso que hacen los usuarios de Internet.
En esta interfaz Web se podría visualizar gran cantidad de información, es decir toda la información guardada en la base de datos, pero sólo vamos a ofrecer información básica, tales como cuales son la aplicaciones más utilizadas, es decir, qué puertos son los más usados, y una gráfica con el tráfico saliente y entrante de cada usuario de la red (una gráfica por cada IP).

1.2 FASES DEL PROYECTO

Las fases en la elaboración del proyecto han sido las siguientes:
-Familiarización con la captura de paquetes, estructura de éstos, los protocolos de Internet, es decir, el proceso de captura de paquetes y su análisis.

-Búsqueda de algún programa de monitorización de redes, en este caso se ha elegido Ntop [1] y probar su funcionamiento en un entorno con mucho trafico, es decir, con velocidades de 1 Gigabit. Ésto incluye el familiarizarse también con aplicaciones de inyección de tráfico, por ejemplo Iperf [5]. También incluye la instalación de estas aplicaciones y la realización de alguna estadística mediante pruebas.

-Desarrollo de una aplicación que permita controlar el uso que hacen los usuarios de Internet.

-Elaboración del interfaz gráfico que permita visualizar los resultados obtenidos por la aplicación que monitoriza la red.

1.3 EQUIPOS NECESARIOS Y MAQUETA

Los equipos necesarios para desarrollar el proyecto son:

-Ordenadores con un buen procesador que permitan inyectar tráfico a altas velocidades y que tengan una tarjeta de red [10] de 1 Gigabit, para poder capturar el tráfico inyectado. En este caso se necesitan 3 máquinas, una que actúa como sniffer (en la que está instalada mi aplicación),otra para inyectar tráfico y otra para recibir tráfico.

-Un switch con puertos de un Gigabit, que permita redirigir el trafico correctamente a esa velocidad y que soporte port mirroring. En la figura 1 se puede ver la maqueta resultante.

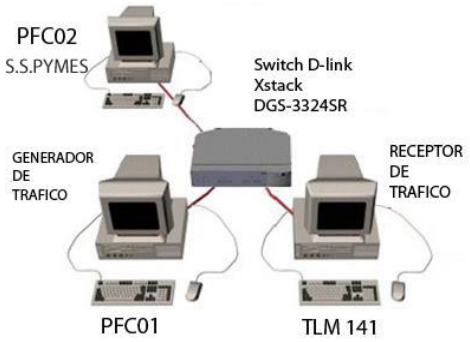


Figura 1 Maqueta del proyecto

La máquina TLM 141 con Linux 2.6.24-24-server Ubuntu, con procesador Intel(R) Pentium(R) 4 CPU 3.00GHz,tiene 80.0GB de disco duro. Posee una tarjeta de red [10] D-Link System Inc Gigabit Ethernet.
IP: 192.168.0.1 máscara 255.255.0.0

La máquina PFC01 IVAN con Linux 2.6.22.9-61.fc6 Fedora 6 , con procesador Intel(R) Pentium(R) 4 CPU 3.40GHz, tiene 160.0 GB de disco duro. Posee una tarjeta de red [10] Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet PCI Express.
IP: 192.168.0.2/16

La máquina PFC2 OSCAR con Linux 2.6.22.9-61.fc6 Fedora 6, con procesador Intel(R) Pentium(R) 4 CPU 3.40GHz, tiene 160.0 GB de disco duro. Tiene una tarjeta de red [10] Broadcom Corporation NetXtreme BCM5751 Gigabit Ethernet PCI Express.
IP: 192.168.0.3/16

Switch D-Link xStack DGS-3324SR, con 24 puertos 10/100/1000BASE-T Gigabit, 2 puertos de 10 Gigabit, 4 puertos COMBO SFP y 1 puerto consola.
Las dimensiones son 441 x 207 x 44mm (ancho, profundidad, altura) y pesa 3.2 KG.
También proporciona una variedad de opciones avanzadas: control de flujo, capacidad duplex, encaminamiento, conmutación Layer 3, soporte de DHCP, soporte BOOTP, soporte ARP, soporte VLAN, copia de puertos, activable, apilable, almacenamiento y reenvio.

IP 192.168.0.4/16

2.NTOP

Ntop es un analizador de tráfico de red que muestra el uso de la red. Ntop se basa en la librería libpcap [7] y ha sido escrito en una forma portátil con el fin de ejecutarse en prácticamente todas las plataformas Unix y en Win32 también.

Se ha elegido este programa porque es muy eficiente y ofrece una gran cantidad de información acerca de todo lo que circula por la red, tiene un gran interfaz gráfico con muchos tipos de gráficas, las cuales organizan la información de diferentes formas.

Los usuarios de Ntop pueden usar el navegador Web para navegar a través de Ntop (que actúa como un servidor Web) y ver la información del tráfico y obtener un volcado del estado de la red.
Qué cosas puede realizar Ntop:

- Ordenar el tráfico de red de acuerdo a los protocolos.
- Analizar el tráfico IP y clasificar de acuerdo a la fuente / destino
- Mostrar el tráfico de red según diferentes criterios.
- Mostrar las estadísticas de tráfico.
- Almacenar las estadísticas de tráfico en un disco de forma persistente en formato RRD.
- Identificar la identidad (por ejemplo, dirección de correo electrónico) de los usuarios.
- Mostrar la distribución del tráfico IP entre los diferentes protocolos.

Plataformas:

- Win32 (Win95 y superiores, incluyendo Vista)
- Unix (incluyendo Linux, BSD, Solaris y MacOSX)

La siguiente tabla muestra la información registrada por Ntop para cada ordenador conectado a la red:

Datos enviados /recibidos	El total de tráfico (cantidad en valor y paquetes) generados y recibidos por el host. Clasificados según el protocolo de red (IP,IPX, AppleTalk, etc.) y el protocolo IP (FTP, HTTP, NFS, etc.).
El uso del ancho de banda	Actual, media y el pico (la máxima velocidad) del uso del ancho de banda.
IP multicast	El total de tráfico multicast generado y recibido por el host.
Historia de las sesiones TCP	Las sesiones TCP establecidas actualmente/aceptadas por el host y las asociadas estadísticas de tráfico.
Tráfico UDP	El total de tráfico UDP ordenado por puertos.
Servicios TCP/UDP usados	Lista de servicios basados en IP (ej. puertos abiertos y activos) ofrecidos por el host con la lista de los últimos 5 hosts que los han usado.
Distribución del tŕafico	Tráfico local, tráfico de local a remoto, tráfico de remoto a local.
Distribución del tráfico IP	Tráfico UDP vs. TCP, distribución relativa de los protocolos IP de acuerdo a su nombre de host.
Distribución de los paquetes	El total del número de paquetes ordenados por el tamaño de paquete , unicast vs. broadcast vs. multicast and IP vs. tráfico que no es IP.
Utilización y distribución de protocolos	Distribución del tráfico observado de acuerdo a ambos protocolos fuente/destino (local vs. Remoto).
Flujos de red	Estadísticas de tráfico según el flujo definido por el usuario (el filtro que el usuario haga).

Ntop posee ayuda para detectar algunos problemas de configuración como:
· Uso duplicado de direcciones IP.
· Identificación de ordenadores locales en “modo promíscuo”.
· La configuración errónea de aplicaciones de software, mediante el análisis del tráfico de datos de protocolo.
· Servicio de detección de uso indebido.
· Identificación de hosts que no hacen uso de proxies específicos.
· Protocolo de uso indebido.
· Identificación de hosts que hacen uso innecesario de protocolos.
· Identificación de routers de subredes.
· Utilización de excesivo ancho de banda en la red.

Las figuras 2, 3, 4 y 5 muestran la información que proporciona Ntop en su interfaz Web:

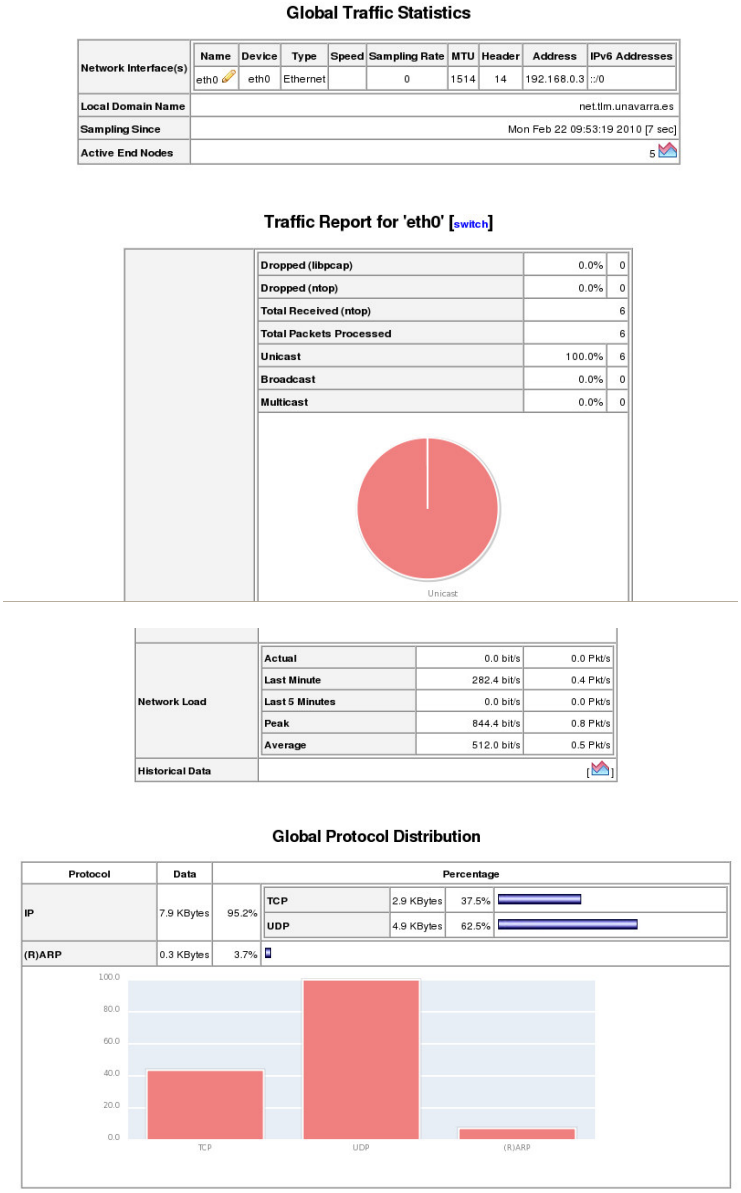
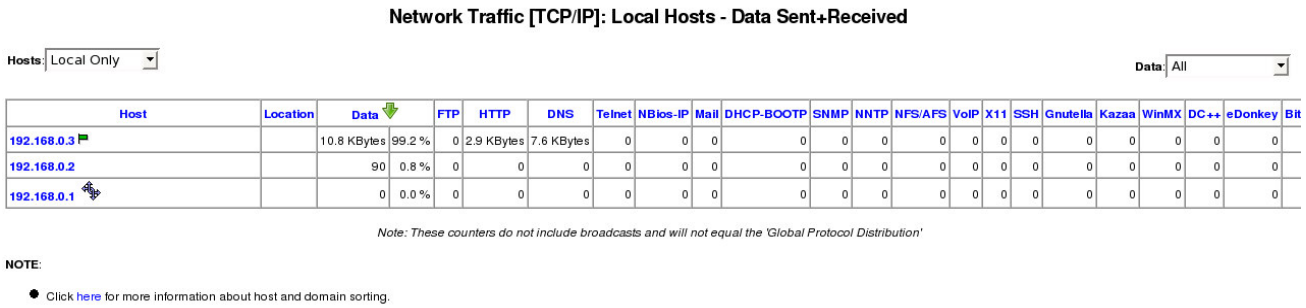


Figura 2 Interfaz principal Ntop

La figura 2 da información de la interfaz por la que Ntop está cogiendo los paquetes IP, los paquetes perdidos, recibidos y proporciona gráficas. Informa de la velocidad de la red, y a qué protocolos pertenece cada paquete.



Network Traffic [TCP/IP]: Local Hosts - Data Sent+Received

Hosts: Local Only — Data: All

Host	Location	Data		FTP	HTTP	DNS	Telnet	NBios-IP	Mail	DHCP-BOOTP	SNMP	NNTP	NFS/AFS	VoIP	X11	SSH	Gnutella	Kazaa	WinMX	DC++	eDonkey	Bit
192.168.0.3		10.8 KBytes	99.2 %	0	2.9 KBytes	7.6 KBytes	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
192.168.0.2		90	0.8 %	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
192.168.0.1		0	0.0 %	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

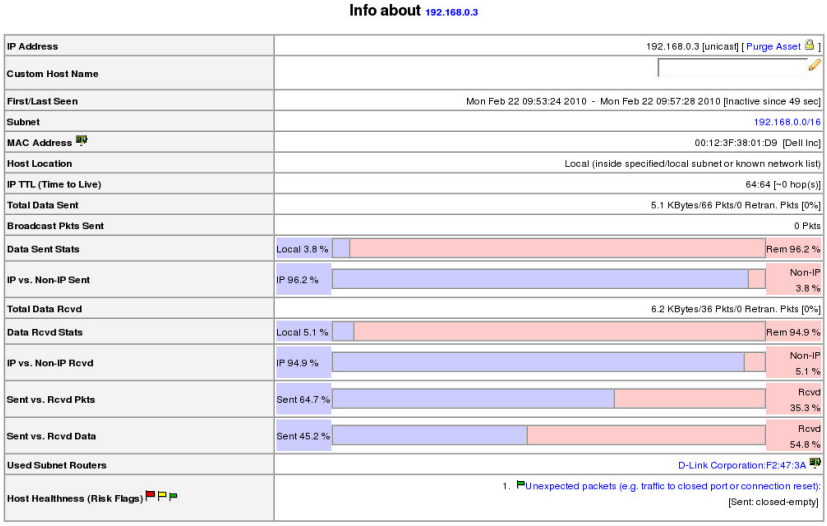
Note: These counters do not include broadcasts and will not equal the 'Global Protocol Distribution'

NOTE:

- Click here for more information about host and domain sorting.

Figura 3 Interfaz por IP Ntop

En la figura 3 se ve información ordenada por IP tales como si ha hecho conexión telnet, ha utilizado el Kazaa, el eDonkey, el Messenger,etc, de todas las IPs que el Ntop ha conseguido algún paquete.

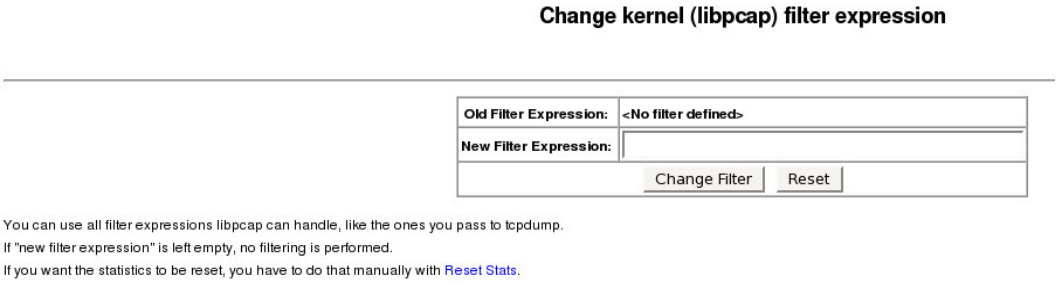


Info about 192.168.0.3

IP Address	192.168.0.3 [unicast] [Purge Asset]
Custom Host Name	
First/Last Seen	Mon Feb 22 09:53:24 2010 - Mon Feb 22 09:57:28 2010 [Inactive since 49 sec]
Subnet	192.168.0.0/16
MAC Address	00:12:3F:38:01:D9 [Dell Inc]
Host Location	Local (inside specified/local subnet or known network list)
IP TTL (Time to Live)	64:64 [~0 hop(s)]
Total Data Sent	5.1 KBytes/66 Pkts/0 Retran. Pkts [0%]
Broadcast Pkts Sent	0 Pkts
Data Sent Stats	Local 3.8 % — Rem 96.2 %
IP vs. Non-IP Sent	IP 96.2 % — Non-IP 3.8 %
Total Data Rcvd	6.2 KBytes/36 Pkts/0 Retran. Pkts [0%]
Data Rcvd Stats	Local 5.1 % — Rem 94.9 %
IP vs. Non-IP Rcvd	IP 94.9 % — Non-IP 5.1 %
Sent vs. Rcvd Pkts	Sent 64.7 % — Rcvd 35.3 %
Sent vs. Rcvd Data	Sent 45.2 % — Rcvd 54.8 %
Used Subnet Routers	D-Link Corporation:F2:47:3A
Host Healthness (Risk Flags)	1. Unexpected packets (e.g. traffic to closed port or connection reset): [Sent: closed-empty]

Figura 4 Información de una IP específica de Ntop

Información específica de una IP (en nuestro caso la 192.168.0.3), como a qué red pertenece, la dirección MAC, total de bytes enviados, total de bytes recibidos, etc.



Change kernel (libpcap) filter expression

Old Filter Expression:	<No filter defined>
New Filter Expression:	

Change Filter | Reset

You can use all filter expressions libpcap can handle, like the ones you pass to tcpdump.
If "new filter expression" is left empty, no filtering is performed.
If you want the statistics to be reset, you have to do that manually with Reset Stats.

Figura 5 Opción de filtrado en Ntop

En la figura 5 podremos hacer unas restricciones a la hora que el Ntop capture los paquetes, estableciendo un filtro, por ejemplo: un determinado protocolo, una IP destino particular, etc. La forma de filtrado es como el tcpdump [8].
En realidad lo que hace es filtrar y descartar todos los paquetes que no se encuentren dentro del filtro, así los ignora y no serán ni procesados ni nos darán información de ellos.
Por ejemplo si hacemos un filtrado por "tcp", y mandamos paquetes UDP, estos serán ignorados y por lo tanto no serán analizados, ya que solo procesará los paquetes que sean de tipo TCP.

2.1 INSTALACIÓN DE NTOP

Primero nos conectamos a su pagina Web oficial www.ntop.org. Aquí nos podemos descargar la ultima versión de esta aplicación la cual es el ntop 3.3.10 (ahora habrá alguna más reciente) . Pinchamos y nos aparecerá una pantalla para descargarlo directamente.
Una vez descargado lo descomprimimos (tar xvzf ntop-3.3.10.tar.gz). Luego entramos en la carpeta descomprimida y habrá un fichero llamado install.txt que te indica como tienes que hacer la instalación. Ahora lo voy a resumir:

En la consola con poner los comandos ./configure, make y make install, sería suficiente para configurar, compilar e instalar este paquete. Lo detallo un poco mas:
Con el comando configure lo que se pretende es chequear que son correctas todas las variables dependientes del sistema utilizadas durante la instalación.

1. 'cd ' al directorio que contiene el código fuente del paquete.
   '. / configure' para configurar el paquete de su sistema (chequea que están instaladas todas las dependencias necesarias durante la compilación). Ésto creará un archivo makefile en cada directorio del paquete y también puede crear uno o varios ficheros .h.
2. Poner 'make' en la consola para compilar el paquete.
3. Opcionalmente poner 'make check' para ejecutar algún test que viene con el paquete.
4. Poner en la consola 'make install' para instalar los programas.
5. Si quiere desinstalarlo 'make uninstall'.

Durante la instalación de este programa se han necesitado una serie de dependencias que en el sistema no estaban ya que es un sistema operativo obsoleto, el Fedora 6.
Instalación Ntop
- cd "directorio de la carpeta ntop-3.3.10"
- ./autogen-sh

fallan las dependencias
Necesita la librería libpcap [7] que se descargó de internet (libpcap-1.0.0) y se ha instalado poniendo en la consola:
- cd "directorio de la carpeta libpcap-1.0.0"
- ./configure
- make
- make install

También necesitaba una versión superior de libevent, mayor que la 1.4.x, se descargó la 1.4.4-stable que es la versión superior más estable y se ha instalado por la consola también:
- cd "directorio de la carpeta libevent-1.4.4-stable"
- ./configure
- make

- make install

También me pedía que tendría instalado el rrdtool y se descargó la versión rrdtool-1.4.2 y al igual que todo instalada en la consola;

- cd "directorio de la carpeta rrdtool-1.4.2"
- ./configure
- make
- make install

Error que no encuentra el directorio 'm4'

- cd "directorio de la carpeta ntop-3.3.10"
- mkdir m4

Pero ahora para compilar Ntop se necesita hacer
./autogen.sh --prefix=/root/Desktop/ntop/ --with-rrd-home=/opt/rrdtool-1.4.2/ (para indicarle que lo instale en la carpeta ntop y decirle que busque el rrdtool en la carpeta donde están todos los archivos)
...autogen.sh done

Después make
Luego makefile

Para la ejecución de Ntop por primera vez: /root/Desktop/ntop/bin/ntop -P /root/Desktop/ntop/share/ntop -u root -A
Después pide que se le introduzca una contraseña :OSCAR.
Y así se ha creado un usuario para el Ntop.
Luego la ejecución normal se hace igual pero con las distintas opciones que tiene este programa, por ejemplo: /root/Desktop/ntop/bin/ntop -P /root/Desktop/ntop/share/ntop -u root -w 3000 (usuario root, corre en el puerto 3000)
Para ver a través de su interfaz Web todas las estadísticas: http://localhost:3000/

2.2 INYECCIÓN DE TRÁFICO (IPERF)

Otra aplicación utilizada en el proyecto es Iperf. Se ha elegido porque es un programa sencillo de utilizar [9], usando un cliente y un servidor, para inyectar gran cantidad de tráfico en la red.
Iperf es una herramienta de prueba de uso de red que puede crear flujo de datos UDP y TCP, y medir el rendimiento de una red gracias a estos flujos. Es una herramienta moderna para la medición de rendimiento de la red escrito en C++.
Permite al usuario ajustar diversos parámetros que pueden ser utilizados para las pruebas de una red, o alternativamente para optimizar la información o el ajuste de una red. Iperf tiene la funcionalidad de cliente y servidor y puede medir el rendimiento entre los dos extremos, ya sea unidireccional o bi-direccional. Es un software de código abierto y funciona en varias plataformas, incluyendo Linux, Unix y Windows.
Cuando se usa para pruebas de capacidad UDP, Iperf permite al usuario especificar el tamaño de datagrama y proporciona los resultados para el rendimiento de datagramas y la pérdida de paquetes.
Cuando se usa para pruebas de la capacidad de TCP, Iperf mide el rendimiento de la carga útil.
Iperf proporciona una salida que contiene un informe los tiempos y la cantidad de datos transferidos y el rendimiento medido.
Se trata de una herramienta de uso muy común que se puede ejecutar sobre cualquier red y salida de las mediciones de rendimiento estandarizados. Por lo tanto, puede ser utilizado para la comparación de cableado y equipos de redes inalámbricas y tecnologías de una manera imparcial.

La forma de utilizar Iperf es creando un cliente y un servidor y mandándose información. Para crearlos se ejecutan estos comandos:
servidor -> iperf -s -u -P 0 -i 1 -p 5005 -l 64 -f M -t 100

-s : servidor.
-u: paquete UPD.
-P: número de conexiones de manejar por el servidor antes de cerrar,con 0 acepta conexiones para siempre
-i: intervalo de tiempo en segundos en que nos muestra el informe con datos de la inyección.
-p: el puerto que utiliza el servidor.
-l: la longitud del paquete.
-f: la unidad de velocidad en la que se va mostrar la información por pantalla.
-t: el tiempo de duración en segundos del servidor.

cliente -> iperf -c <ipservidor> -u -P 5 -i 1 -p 5005 -l 64 -f M -b 100M -t 100 -L 5001 -T 1

-c: cliente.
-<ipservidor> :IP del servidor al que te quieras conectar.
-u :paquete UDP.
-P: número de subprocesos cliente que se ejecutarán en paralelo.
-i: el intervalo de tiempo en segundos en que nos muestra el informe con datos de la inyección.
-p: el puerto al que conectarse.
-l: la longitud del paquete a enviar.
-f: la unidad de velocidad en la que se va mostrar la información por pantalla.
-b: el ancho de banda, la unidad de velocidad por segundo a la que va a mandar información.
-t: el tiempo de duración en segundos del servidor.
-L: puerto para recibir pruebas bidireccionales.
-T: time-to-live.

2.3 EVALUACIÓN NTOP

Para verificar el programa de Ntop lo que se ha hecho es hacer una prueba que consiste en ver según el tamaño de paquete y la velocidad de la red, el porcentaje de pérdidas de paquetes que sufre este programa.
Se puede visualizar en la figura 6, en cuyo eje vertical se encuentra el porcentaje de perdidas de paquetes, en el eje horizontal el número de paquetes por segundo, y cada línea es el tamaño de paquete elegido.

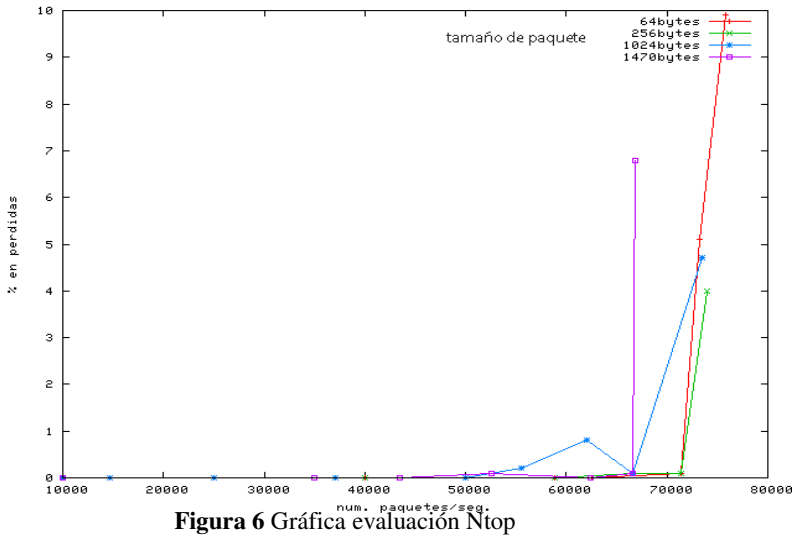


Figura 6 Gráfica evaluación Ntop

Primero se realizó esta prueba cargando el interfaz Web del programa de vez en cuando durante las pruebas para ir viendo como iba perdiendo gradualmente paquetes y eso provocaba alteraciones en los resultados, es decir, que si realizaba una prueba sin abrir la página Web y la volvía a repetir abriendo el navegador varias veces el resultado no era el mismo, luego se llegó a la conclusión de que el cargar el interfaz Web podría tener una perdida de eficiencia en el programa a altas velocidades.
Y ya luego se realizó esta prueba sin cargar la página Web y los resultados son los de la gráfica anterior, vamos que son los resultados propios de Ntop , sin ser afectado por nada.

Para ver que realmente la carga del interfaz Web afecta al rendimiento de Ntop se hizo otra prueba.
Abrir el navegador Web donde aparecen las estadísticas de Ntop sin realizar prácticamente ninguna estadística, es decir, nada más iniciar el programa, tarda 1.5 segundos en cargarse. Ahora se va a hacer una prueba de la carga de la página Web mientras está analizando tráfico y se verá como afecta también a la hora de la pérdida de paquetes.
Ejemplo:
Primero se ha realizado la prueba inyectando tráfico en la red durante 100 segundos y sin actualizar la página Web salen estos resultados: utilizando un tamaño de paquete de 1024 bytes, Ntop recibe 50000 paquetes por segundo y pierde 9 paquetes por segundo, casi nada.
Sin embargo, al volver a hacer la misma prueba, pero actualizando la página Web una vez en la mitad de la prueba (a los 50 segundos) salen estos resultados: Ntop recibe prácticamente los mismos paquetes 49220 por segundo pero ahora pierde 776 paquetes por segundo, es decir pierde el 1.6%, y tarda en cargarse la página ahora 2.5 segundos.
Llegamos a la conclusión de que el procesamiento de todas las estadísticas de Ntop, y sacarlas por pantalla mediante gráficas y demás es muy costoso para él y afecta bastante a su rendimiento.

En la figura 7 hay una gráfica que muestra la relación entre el tamaño de paquete y la velocidad, cuando la aplicación se encuentra en el límite de paquetes procesados, es decir, cuando no se pierden paquetes a la velocidad máxima.

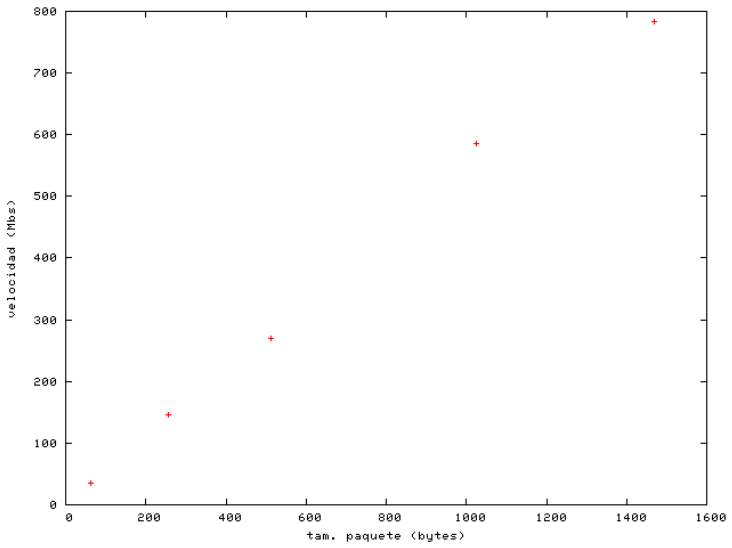


Figura 7 Gráfica velocidad máxima sin perdidas según el tamaño de paquete

Como conclusión se puede ver que la gráfica es prácticamente lineal, es decir a medida que el tamaño de paquete va aumentando la velocidad también. Por lo cual podemos determinar que lo que más le cuesta al procesador es colocar el paquete, dando igual el tamaño que tenga, y por eso a mayor tamaño de paquete, mayor cantidad de datos y por lo tanto mayor velocidad.
He aquí otra gráfica (figura 8)que muestra la relación entre el tamaño de paquete y el numero de paquetes por segundo, en el momento que la aplicación está al límite de procesado:

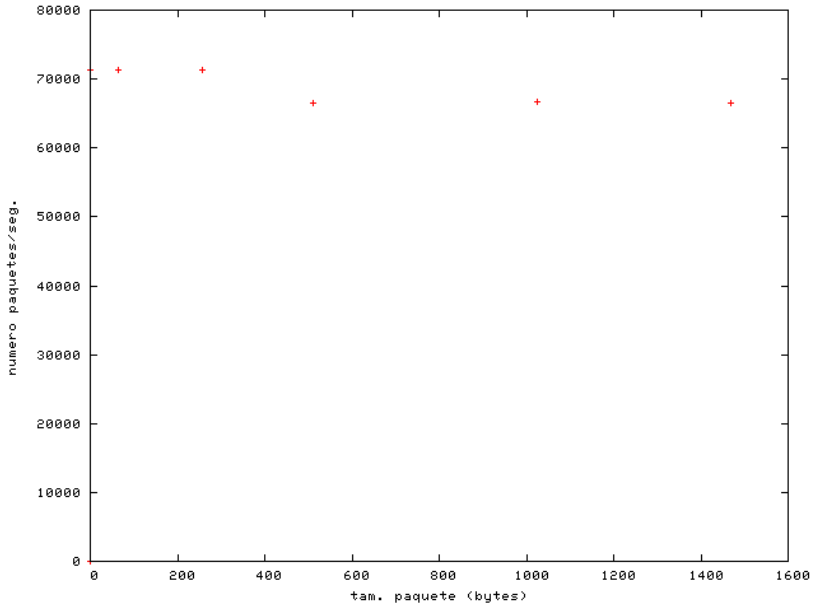


Figura 8 Número de paquetes procesados por segundo según el tamaño de paquete

La conclusión que se puede sacar es que sea cual sea el tamaño del paquete, la aplicación va a poder procesar más o menos el mismo numero de paquetes, luego lo que va a importar es el número de paquetes por segundo, no la velocidad.

3. DISEÑO DEL SISTEMA DE MONITORIZACIÓN

La aplicación que se ha conseguido desarrollar es el S.S. PYMES (Sistema de Supervisión de PYMES).
El objetivo de este prototipo es la realización de un programa que sea capaz de capturar los paquetes de la red que pasen por su tarjeta de red [10], los analice y los guarde en una base de datos para posteriormente realizar unas estadísticas que serán publicadas en una página Web.
Este programa se divide en cuatro módulos y más tarde hablaremos de cada uno de ellos con más detalle:
-Captura: consiste en la captura de los paquetes de red desde la tarjeta de red.
-Procesamiento: consiste en el análisis de los paquetes y la obtención de la información necesaria.
-Almacenamiento: guardar la información necesaria en una base de datos.
-Interfaz de usuario: volcar la información de la BD a una página Web.
Aquí se muestra un diagrama (figura 9) con el estructura del programa para que quede más claro:

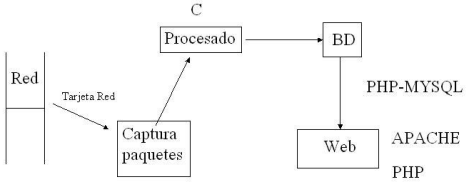


FIGURA 9 Módulos del programa S.S.PYMES

4. MÓDULO DE CAPTURA

Este módulo ha sido realizado mediante un programa en C [17]. Se ha escogido el lenguaje C ya que se trata de un lenguaje débilmente tipificado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Por lo tanto me permite acceder a los paquetes que se encuentran en mi tarjeta de red de una manera sencilla, para así analizarlos.
Gracias a la librería libpcap [7] somos capaces de acceder a la tarjeta de red y obtener información tanto de los paquetes, como de la red. Dispone de un conjunto de funciones [15] muy interesantes
Una parte del código fue obtener los datos de la red y la máscara del interfaz de red aunque éstos no serán mostrados por pantalla porque ralentizan el programa.

```
//printf("Obteniendo Tarjeta de red [");

if ((dev = pcap_lookupdev(errbuf))==NULL){ //obtener la tarjeta de red
        printf("Error]-> %s\n", errbuf);
        exit(-1);
}
//printf("OK] \n-> %s\n",dev);

//printf("Obtener red y mascara [");

if ((ret = pcap_lookupnet(dev, &netp, &maskp, errbuf))==-1){ //obtener mascara y red
        printf("Error]-> %s\n", errbuf);
        exit(-2);
}

addr.s_addr = netp; //traducimos la direccion de red a algo legible
if ((net = inet_ntoa(addr))==NULL){
        printf("Error]->  Red -> inet_ntoa\n");
        exit(-3);
}

//printf("OK] \n-> Red: %s\n",net); //sacar por pantalla la red

addr.s_addr = maskp; //traducimos la mascara a algo legible
if ((mask = inet_ntoa(addr))==NULL){
        printf("[Error]->  Mascara de Red -> inet_ntoa\n");
        exit(-4);
}
//printf("-> Mascara de Red: %s\n", mask); //sacar por pantalla la mascara
```

Figura 10 Código referido a la obtención de datos de la red

Otra es la obtención del nombre del host y la apertura del primer interfaz de red disponible para poder capturar a través de él.

```
//printf("Obteniendo el nombre del host [");
if (gethostname(nhost,sizeof(nhost)) < 0) {
        printf("Error] consultado el hostname.\n");
        exit(-6);
}
//printf("OK] \n-> El nombre del host es: %s\n",nhost);


if ((descr=pcap_open_live(dev,BUFSIZ,1,20,errbuf))==NULL){
        printf("Error en open live: %s\n",errbuf);
        exit(-8);
}
```

Figura 11 Código referido a la obtención del nombre de host y la apertura del interfaz

La parte de código que consiste en la captura de los paquetes se trata de una función que cada vez que llega un paquete a la tarjeta de red. Se ejecuta y es la encargada de llamar el siguiente módulo, el de procesamiento. La función pcap_loop estará en un bucle infinito esperando a que lleguen paquetes y cada vez que llegue uno, ejecutará la función llegada. Se le pasa el descriptor del interfaz de red que anteriormente se había abierto, un 0 para que capture infinitamente (poniendo un número, el programa termina cuando se capturan ese número de paquetes que se ha puesto) y la función que se ejecutará.

```
pcap_loop(descr,0,llegada,NULL);

//funcion que se ejecuta cada vez que la tarjeta de red captura un paquete
void llegada(u_char *useless,const struct pcap_pkthdr* pkthdr,const u_char* packet)
```

Figura 12 Código del bucle de la captura de paquetes

También hay una posibilidad de establecer un filtro de captura, por ejemplo, si se pone como filtro "udp" sólo capturará aquellos paquetes cuyo tipo es UDP y los demás los ignorará, sólo pasarán a la BD aquellos que cumplan el filtro.

```
if (argc>1 ){
        printf("*** Opcion de Filtrado Activa ***\n ");
        printf("Compilar Filtro [");

        if ((pcap_compile(descr, &fp, argv[1], 0, netp)) == -1){
                printf("Error]\n Compilando el filtrado %s:\t",argv[1]);
                printf(" %s\n",pcap_geterr(descr));
                exit(-9);
        }
        printf("OK] %s\n",argv[1]);

        printf("Estableciendo Filtro [");

        if ((pcap_setfilter(descr,&fp))==-1){
                printf("Error]\n Filtro: \n");
                printf(" %s\n",pcap_geterr(descr));
                exit(-10);
        }

        printf("OK]\n");
```

Figura 13 Código de filtrado

5. MÓDULO DE PROCESAMIENTO

Este módulo se encarga de analizar cada paquete y de conseguir la información que más nos interesa de cada uno. Se trata de una función que se ejecuta cada vez que un paquete entra a la tarjeta de red. Por ejemplo algunos datos interesantes son la IP origen del paquete, la IP destino, los puertos, los bytes del paquete, qué tipo de paquete es (IP, ARP, TCP, UDP, etc).
La forma de obtención de información de un paquete del tipo TPC es ésta:

```
//printf(" TCP ");
struct tcphdr *tcpc = (struct tcphdr*)(packet+sizeof(struct ether_header) + (ipc->ip_hl*4));
//printf("PO:%d ", ntohs(tcpc->source));                //Puerto de Origen:
//printf("PD:%d ", ntohs(tcpc->dest));          //Puerto de Destino:
int tmp_len_tcphdr = tcpc->doff*4;
//printf("LC:%d ", tmp_len_tcphdr);
//printf("NS:%d ", ntohs(tcpc->seq));           //Numero de Secuencia:
//printf("NA:%d ", ntohs(tcpc->ack_seq));               //Numero de acuse de recibo:
//printf("cwr:%d ", tcpc->res2&0x01);
//printf("ece:%d ", tcpc->res2&0x02);
//printf("urg:%d ", tcpc->urg);
//printf("ack:%d ", tcpc->ack);
//printf("psh:%d ", tcpc->psh);
//printf("rst:%d ", tcpc->rst);
//printf("syn:%d ", tcpc->syn);
//printf("fin:%d ", tcpc->fin);
//printf("WI:%d ", tcpc->window);
//printf("CS:%d ", tcpc->check);
//printf("UR:%d\n", tcpc->urg_ptr);
```

Figura 14 Obtención de información de un paquete de tipo TCP

6.ALMACENAMIENTO DE LA INFORMACIÓN

Para el programa S.S.PYMES se necesita almacenar una buena cantidad de información: IPs, numero de paquetes de cada IP, número de bytes mandados o recibidos de una IP, puerto usado, etc.
Para ello se ha planteado de guardarla o en ficheros o en bases de datos. Por tanto se ha realizado un estudio para comprobar si la utilización de bases de datos es suficiente para este caso, es decir, dentro de una pequeña empresa en la que su velocidad de red no será muy elevada.
Este estudio consiste en ver qué cantidad de transacciones por segundo es capaz de hacer dicha base de datos: al actualizar datos, al insertarlos , al eliminarlos y al hacer una selección de éstos.
Se ha hecho un programa por cada tipo de transacción (actualización, inserción, selección y borrado) ejecutando una sola sentencia por vez y luego se han encontrado unas mejoras, tales como agrupar y ejecutar varias sentencias por transacción y veremos que los resultados mejoran.

Empecemos por las transacciones sin agrupar. Un ejemplo de inserción (figura 15):

```
//printf("COMIENZA LA PRUEBA PARA AGRUPAR SENTENCIAS Y INSERTAR DATOS\n");
srand(time(NULL));
for(i=1;i<=100000;i=i+1){
        azar = (int) rand () % (1000+1);
        sprintf(consulta,"insert into prueba (ip,contador) values (%d,%d)",i,azar);
        if (mysql_real_query(conn,consulta,strlen(consulta))) {
                fprintf(stderr, "error insercion: %s\n", mysql_error(conn));
                return(0);
        }
}
```

Figura 15 Código para la prueba de inserción de una sola sentencia

Los resultados son los siguientes:

14.084 inserciones/ seg.
11.904 borrados/ seg.
11.764 actualizaciones/ seg.
10.309 selecciones/ seg.

Agrupando las sentencias de 10 en 10 para ejecutarlas de una sola vez es más o menos lo mas efectivo (es entre 5 y 10 aunque ahora iremos viendo con el estudio que voy a seguir haciendo cuál es el valor óptimo de agrupamiento para cada transacción, etc) .
Aquí un ejemplo de la inserción anterior pero agrupadas de 10 en 10:

```
                //printf("COMIENZA LA PRUEBA PARA AGRUPAR SENTENCIAS Y INSERTAR DATOS\n");
                srand(time(NULL));
                for(i=1;i<=100000;i=i+10){
                        azar = (int) rand () % (1000+1);
                        sprintf(consulta,
"insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);\
insert into prueba (ip,contador) values (%d,%d);",i,azar,i+1,azar,i+2,azar,i+3,azar,i+4,azar,i+5,azar,i+6,azar,i+7,azar,i+8,azar,i+9,azar);
                        if (mysql_real_query(conn,consulta,strlen(consulta))) {
                                fprintf(stderr, "error insercion: %s\n", mysql_error(conn));
                                return(0);
                        }
                        for(j=1;j<=10;j++){
                                mysql_next_result(conn);
                        }
                }
```

Figura 16 Código para la prueba de inserción de 10 sentencias a la vez

Los resultados son los siguientes:

20.408 inserciones/ seg.
16.393 borrados/ seg.
16.129 actualizaciones/ seg.
15.625 selecciones/ seg.

En principio, la transacción que más veces se va a ejecutar es la actualización. Por tanto 16000 transacciones, a 200 bytes cada paquete (tamaño medio del paquete) son 25.6 Mbps.
16000*200 bytes*8= 25.6 Mbps
Se puede dar cuenta que la utilización de una BD es suficiente para nuestra problemática. Ahora nos hemos centrado en encontrar optimizaciones:
-El agrupamiento de sentencias anterior
-La utilización de un socket Unix en vez de uno TCP/IP

```
/* Connect to database */
if (!mysql_real_connect(conn, server,
  user, password, database, 0, "/var/lib/mysql/mysql.sock",CLIENT_MULTI_STATEMENTS)) {
        fprintf(stderr, "%s\n", mysql_error(conn));
        return(0);
}
```

Figura 17 Forma de conexión a la BD

Al conectarnos a la BD [16], se le dice que lo haga a través del socket Unix, mucho más rápido [19] que uno TCP/IP, y además se le pone la opción CLIENT_MULTI_STATEMENTS (para soportar la agrupación de sentencias).
La gráfica que resulta al recoger todos los datos es ésta:

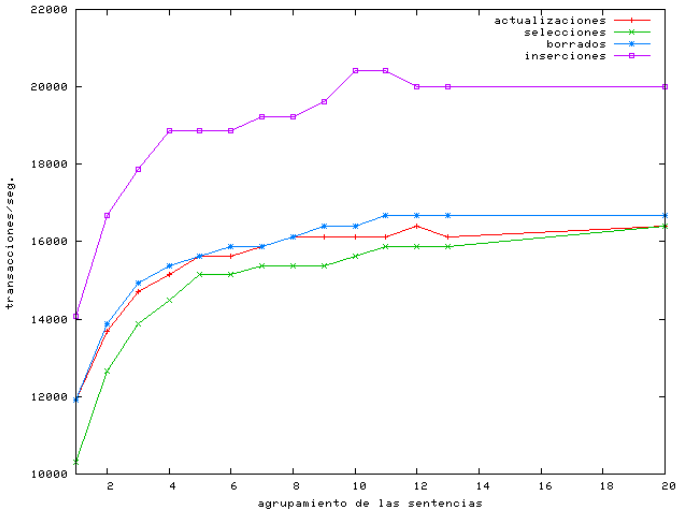


Figura 18 Prueba uso ficheros vs BD

En el eje x aparecen el numero de sentencias agrupadas (de 1 en 1, de 2 en 2, etc, y de 20 en 20) y en el eje y, las transacciones por segundo que es capaz de realizar la base de datos (actualizaciones, inserciones, borrados y selecciones).

En realidad la transacción que más nos interesa es la de actualización, ya que la mayoría de sentencias que creemos que se ejecutarán en nuestra base de datos serán de este tipo.
Como vemos nuestra base de datos va a poder realizar unas 16000 actualizaciones por segundo. Lo cual parece ser suficiente el utilizar una base de datos para poder guardar la información en nuestra pequeña empresa (falta a qué velocidad es capaz de procesar como máximo..).
En cuanto a las selecciones, la primera vez que se realizaron las pruebas daban unos resultados con unos resultados muy bajos, es decir se realizaban pocas transacciones por segundo, que se planteó que podía ser por la asignación de memoria que realiza internamente , entonces añadí una función para liberar memoria cada vez que haga una selección y los resultados mejoraron considerablemente. Sin liberar memoria se realizaban unas 14.000 transacciones por segundo y al liberar realiza unas 15.500.

Como conclusión, decimos que nos resultará suficiente usar las bases de datos como mecanismo para el almacenamiento de información.

Pero a la hora de realizar la implementación aparecieron una serie de problemáticas. Una de ellas es que la sentencia más utilizada es la selección, por ejemplo para ver si esa una IP está en la BD y en función de eso actualizar una fila o insertarla. Y debido a la estructura del programa tampoco se va a poder agrupar varias sentencias y ejecutarlas a la vez, ya que las inserciones y actualizaciones que hay que ejecutar dependen de una consulta y en función de ésta se ejecutará una u otra, por lo tanto los resultados del programa serán un poco peores a los sacados anteriormente.

Ahora se va a explicar la parte de conexión a la base de datos y la organización de ésta.
Era necesario encontrar la manera de conectarse a una base de datos desde C, para guardar esa información, y se hace usando la librería MySQL que proporciona funciones para la comunicación entre C y la base de datos. Se puede ver la parte de código para conectarse a la BD en la figura 17 de la página 17.

En cuanto a la estructura de la BD era necesario realizarla bien para poder guardar el tipo de información que más nos interesaba: IP, puerto, protocolo, etc, de una forma fácil, para luego poder recuperar esa información de manera sencilla.
Ahora se va a proceder a explicar el esquema de la BD.
Va a estar formada de 3 tablas, vistas en las figuras 19, 20 y 21:

-Datos : con toda la información de cada IP, puertos, bytes,etc.

```
mysql> describe datos;
+--------------------+----------+------+-----+---------+-------+
| Field              | Type     | Null | Key | Default | Extra |
+--------------------+----------+------+-----+---------+-------+
| segundo            | int(6)   | NO   | PRI | 0       |       |
| ip                 | char(40) | NO   | PRI |         |       |
| protocolo          | char(20) | NO   | PRI |         |       |
| o_d                | char(1)  | NO   | PRI |         |       |
| puerto_origen      | int(6)   | NO   | PRI | 0       |       |
| puerto_destino     | int(6)   | NO   | PRI | 0       |       |
| num_paq_recibidos  | int(8)   | YES  |     | NULL    |       |
| num_paq_enviados   | int(8)   | YES  |     | NULL    |       |
| bytes_recibidos    | int(12)  | YES  |     | NULL    |       |
| bytes_enviados     | int(12)  | YES  |     | NULL    |       |
+--------------------+----------+------+-----+---------+-------+
10 rows in set (0.05 sec)
```

Figura 19 Tabla datos

-Usuarios: será un mapeo entre IPs y nombre de usuario de la empresa

```
mysql> describe usuarios;
+---------+----------+------+-----+---------+-------+
| Field   | Type     | Null | Key | Default | Extra |
+---------+----------+------+-----+---------+-------+
| ip      | char(25) | NO   | PRI |         |       |
| usuario | char(40) | NO   | PRI |         |       |
+---------+----------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Figura 20 Tabla usuarios

-Aplicaciones: será un mapeo entre puerto y aplicación a la que corresponde

```
mysql> describe aplicaciones;
+------------+----------+------+-----+---------+-------+
| Field      | Type     | Null | Key | Default | Extra |
+------------+----------+------+-----+---------+-------+
| puerto     | int(6)   | NO   | PRI | 0       |       |
| aplicacion | char(40) | NO   | PRI |         |       |
+------------+----------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```

Figura 21 Tabla aplicaciones

7. INTERFAZ DE USUARIO

Para poder volcar esos datos de nuestra base de datos en una página Web, se ha utilizado el lenguaje PHP [4], un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas Web dinámicas. A través el cual habrá que conectarse a la base de datos [18] y obtener la información que interese para mostrarla. En nuestro caso se realizará una interfaz que ofrezca información básica, tales como las aplicaciones más utilizadas, una gráfica para cada IP, mostrando el tráfico entrante y saliente, etc, pero se puede sacar mucha más información y de muchas maneras diferentes. Para ello es necesario usar la aplicación MySQL [3], el servidor Apache [2] , y PHP. En el PC estaban ya instaladas la versión 5.0.27 de MySQL, la 2.2.6 de Apache y la 5.1.6 de PHP. Sin embargo no se podía acceder desde un script en PHP a la base de datos de MySQL y todo era debido que al compilar cualquier versión superior o igual a 5 de PHP desde las fuentes, no viene precompilado el soporte para MySQL. Entonces se ha tenido que instalar una serie de paquetes necesarios para ello, desde una página Web donde hay todo lo necesario [11]:
Era necesario el paquete httpd-devel-2.2.6-1.fc6.i386 (herramientas para el desarrollo del servidor HTTP apache), pero tenía unas dependencias que no tenía instaladas, así que primero se instaló las dependencias.
Primero el paquete apr-util-devel-1.2.8-1.fc6.i386.rpm:
-cd al directorio donde está el paquete.
-rpm -i apr-util-devel-1.2.8-1.fc6.i386.rpm
Segundo el paquete apr-devel-1.2.7-10.i386.rpm:
-cd al directorio donde está el paquete.
-rpm -i apr-devel-1.2.7-10.i386.rpm
Finalmente el paquete httpd-devel-2.2.6-1.fc6.i386.rpm:
-cd al directorio donde está el paquete.
-rpm -i httpd-devel-2.2.6-1.fc6.i386.rpm

También se necesitaba el paquete mysql-devel-5.1.49-1.fc6.remi.i386.rpm (archivos para el desarrollo de aplicaciones MySql):
-cd al directorio donde está el paquete.
-rpm -i mysql-devel-5.1.49-1.fc6.remi.i386.rpm

Ya después de tener todos los paquetes necesarios instalados sólo faltaba volver a configurar e instalar Php-5.1.6, pero era necesario indicarle una ruta donde se encuentran todos los archivos MySQL [3] y también otra ruta para indicarle donde está el archivo apxs. Para ello nos descargamos el paquete de Php-5-1-6 y procedimos a instalarlo:

-cd al directorio donde está el paquete de Php-5.1.6 ya descomprimido.
-./configure –with-apx2=/usr/sbin/apxs –with-mysql=/root/Desktop/mysql-connector-c-6.0.2-linux-glibc2.3-x86-32bit
-make
-make install

Y ya de esta manera está instalado todo para poder realizar el interfaz Web. Sólo faltaba lanzar los servicios, para lanzar apache basta con poner en la consola :
/usr/sbin/apachectl -f /etc/httpd/conf/httpd.conf -k start
y dejar en el directorio /var/lib/mysql/ los archivos .php, que es donde se ha configurado el apache.
También hay que lanzar MySQL creando el socket e indicándole donde está la bd tal que así:
/usr/libexec/mysqld -u root --datadir=/var/lib/mysql/ --socket=/var/lib/mysql/mysql.sock &
Y ya tendríamos todo funcionando.
Entonces lo primero a realizar fue conectarme a la base de datos, con el usuario y contraseña y usando el socket Unix de MySQL. Luego ya ir sacando los datos que interesan de la base de datos, tales como dos tablas, una con la información de los 10 puertos origen más usados y otra con los 10 puertos destino más usados y cada una de las IPs que mandan o reciben paquetes.
Finalmente se han hecho un conjunto de ficheros .php para conseguir la aplicación:

-index.php: la página inicial con 3 links a diferentes estadísticas.
-top_origen.php: muestra los 10 puertos generadores con más tráfico.
-top_destino.php: muestra los 10 puertos receptores con más tráfico.
-maquinas.php: da información de qué maquinas de la red se ha obtenido algún paquete, y hay 2 links para ver más detalladamente el tráfico de esta IP en el tiempo.
-procesa.php: muestra una gráfica con el número de paquetes por segundo del tráfico tanto entrante como saliente de una IP.
-procesa2.php: muestra los bits por segundo del tráfico tanto entrante como saliente de una IP.

Para poder conseguir esas gráficas se ha utilizado el programa Gnuplot [20].Es un programa muy flexible para generar gráficas de funciones y datos. Este programa es compatible con los sistemas operativos más populares (Linux, UNIX, Windows, Mac OS X,etc). El origen de gnuplot data de 1986.
Gnuplot puede producir sus resultados directamente en pantalla, así como en multitud de formatos de imagen, como PNG, EPS, SVG, JPEG, etc. Se puede usar interactivamente o en modo por lotes (batch), usando scripts. Este programa tiene gran base de usuarios y está convenientemente mantenido por sus desarrolladores. Existe una buena cantidad de ayuda en Internet, aunque gran parte está en inglés.
Gnuplot es la herramienta de dibujo de gráficas del programa GNU Octave.
Este programa se distribuye bajo una licencia de software libre que permite la copia y modificación de su código fuente. Sin embargo, las modificaciones sólo se pueden distribuir en forma de parches, por lo que no es compatible con la licencia GPL.
La figura 22 muestra la parte de código en .php correspondiente a la información de las máquinas que hay en la red (maquinas.php), de las cuales se ha capturado algún paquete.

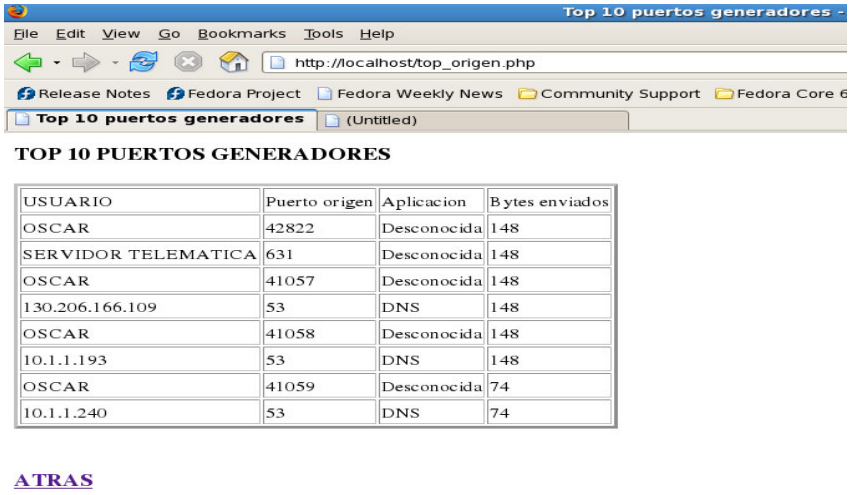
```
<html>
<title>
Máquinas de la red
</title>
<body>
<h3>MÁQUINAS DE LA RED</h3>
<?php
$conexion = mysql_connect(":/var/lib/mysql/mysql.sock", "root","268y483") or die (mysql_error());
mysql_select_db("test",$conexion);

      $consulta_ips = "SELECT distinct ip FROM datos";
      $ejecuta_ip=mysql_query($consulta_ips,$conexion) or die (mysql_error());

      while($fila_ip=mysql_fetch_array($ejecuta_ip)){
            $ip=$fila_ip['ip'];
            $consulta_usuario = "SELECT ip,usuario FROM usuarios where ip= \"$ip\"";
            $ejecuta_usuario=mysql_query($consulta_usuario,$conexion) or die (mysql_error());
            $num_rows = mysql_num_rows($ejecuta_usuario);
            if ($num_rows==0){
                  echo "<p align=\"left\"><font size=\"4\" face=\"Arial\"> Gráfica $ip <a href=\"procesa.php?ip=$ip\"> Nºpaquetes/seg. </a> / <a href=\"procesa2.php?i
            }
            else {
                  $fila_usu=mysql_fetch_array($ejecuta_usuario);
                  $usuario=$fila_usu['usuario'];
            echo "<p align=\"left\"><font size=\"4\" face=\"Arial\"> Gráfica $usuario <a href=\"procesa.php?usu=$usuario&ip=$ip\"> Nºpaquetes/seg. </a> / <a href=\"proc
            }
      }

mysql_close($conexion);
echo "<br><h3><a href=\"index.php\">ATRAS</a></h3>";
?>
</body>
</html>
```

Figura 22 Código correspondientes a maquinas.php

La página de inicio (figura 23), con tres enlaces, uno a el top 10 de puertos generadores, otro al top 10 de puertos destino y el último a las máquinas de la red.

Figura 23 Página principal del interfaz de S.S.PYMES (index.php)

La figura 24 muestra el primer enlace con una tabla con los puertos generadores mas utilizados, el nombre de la aplicación si es que existe y los bytes enviados.



USUARIO	Puerto origen	Aplicacion	Bytes enviados
OSCAR	42822	Desconocida	148
SERVIDOR TELEMATICA	631	Desconocida	148
OSCAR	41057	Desconocida	148
130.206.166.109	53	DNS	148
OSCAR	41058	Desconocida	148
10.1.1.193	53	DNS	148
OSCAR	41059	Desconocida	74
10.1.1.240	53	DNS	74

Figura 24 Página top 10 puertos generadores (top_origen.php)

El segundo link que aparece en la figura 23 es el mismo que el primero pero de los puertos que más trafico reciben.
El tercer link que se ve en la figura 23 son las máquinas de la red (figura 25).

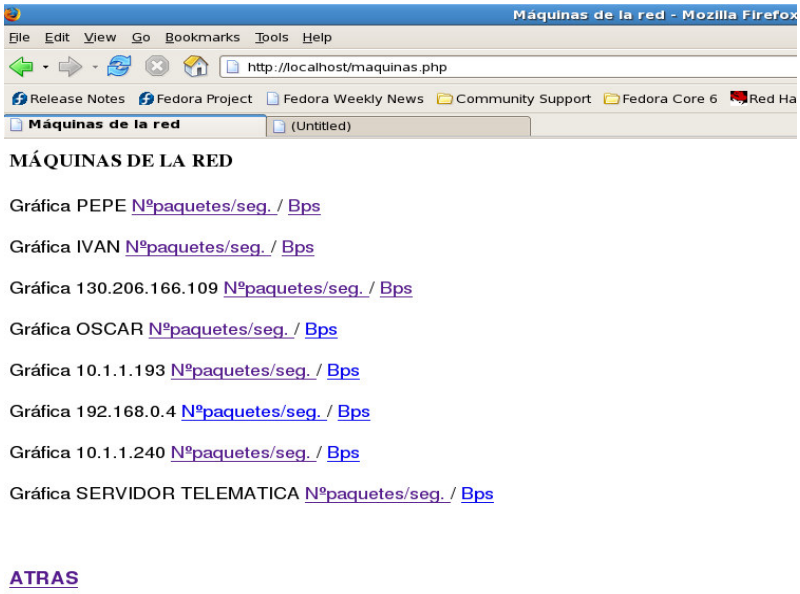


Figura 25 Página Máquinas de la red (maquinas.php)

En esta Web se puede consultar tanto el numero de paquetes por segundo entrantes o salientes de una IP, o los bps enviados y recibidos de dicha IP(figuras 26 y 27 respectivamente).

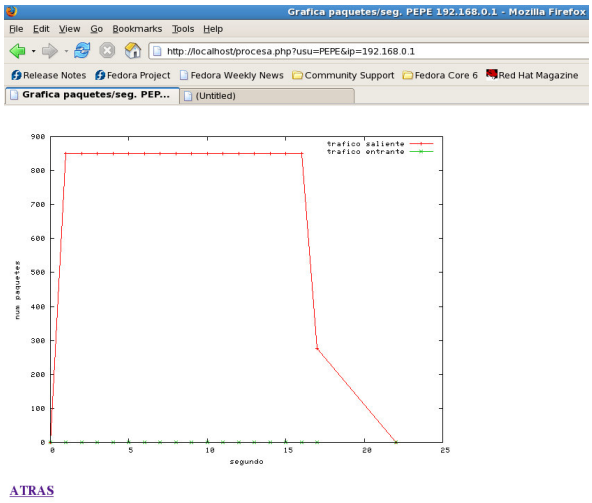


Figura 26 Página número paquetes/seg. Pepe (procesa.php)

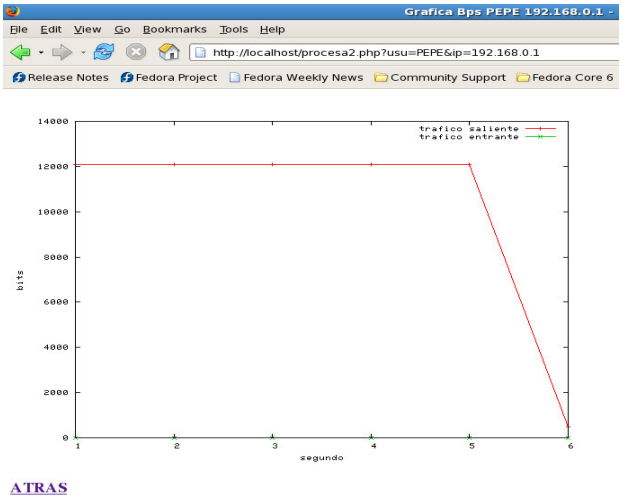


Figura 27 Página bits/seg. Pepe (procesa2.php)

8. EVALUACIÓN S.S.PYMES

Al igual que se realizó con Ntop un estudio, también va a ser realizado sobre el programa implementado S.S.PYMES.
El estudio consistía en ver cuantos números de paquetes es capaz de procesar el prototipo, sin tener pérdidas de paquetes. Aquí está la gráfica (figura 28) en la que se aprecian el porcentaje de pérdidas según el tamaño de paquete y el número de paquetes por segundo:

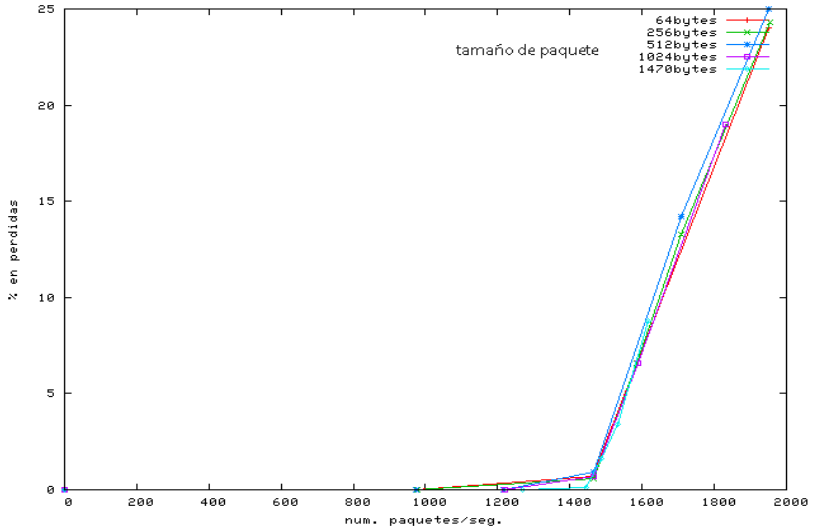


Figura 28 Gráfica evaluación S.S.PYMES

Se llega a la misma conclusión que la que llegamos con Ntop, es indiferente el tamaño de paquete para el programa tenga perdidas, ya que a menor tamaño de paquete a menor velocidad recibirá y viceversa, lo importante es el número de paquetes por segundo que se puede procesar, y como se ve sea cual sea el tamaño del paquete comienza a tener pérdidas en el mismo punto.

Para ver exactamente cuantos paquetes por segundo es capaz de procesar el PC, he hecho un estudio viendo el consumo de la CPU cuando S.S.PYMES está en funcionamiento.

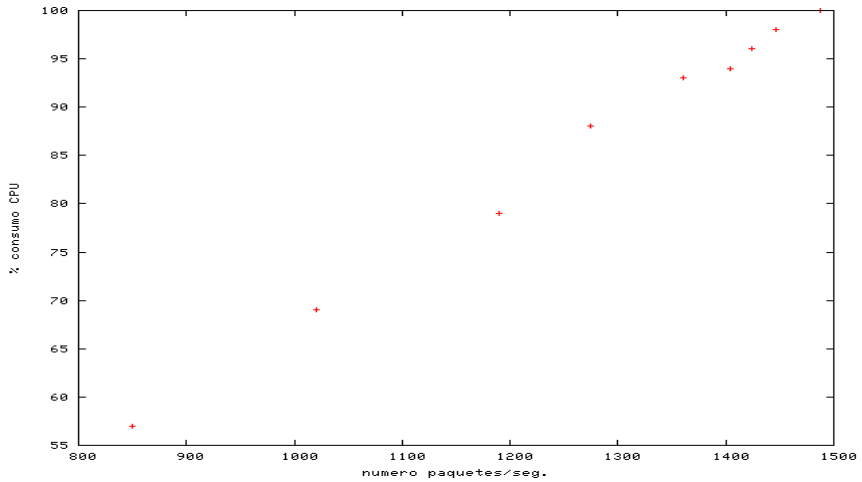


Figura 29 Gráfica evolución consumo CPU según el numero paquetes/seg.

Lo que en realidad sobrecarga la CPU es todo el procesamiento de MySQL, lo que es el procesamiento del programa .c es despreciable. Lo que se puede apreciar que la máquina es capaz de procesar entre 1400 y 1500 paquetes por segundo sin tener pérdidas o muy pocas.

En la figura 30 se ve otra gráfica en la que se ve la relación entre el tamaño de paquete y la velocidad, cuando la aplicación se encuentra al límite de paquetes procesados, es decir, cuando no se pierden paquetes a la máxima velocidad.

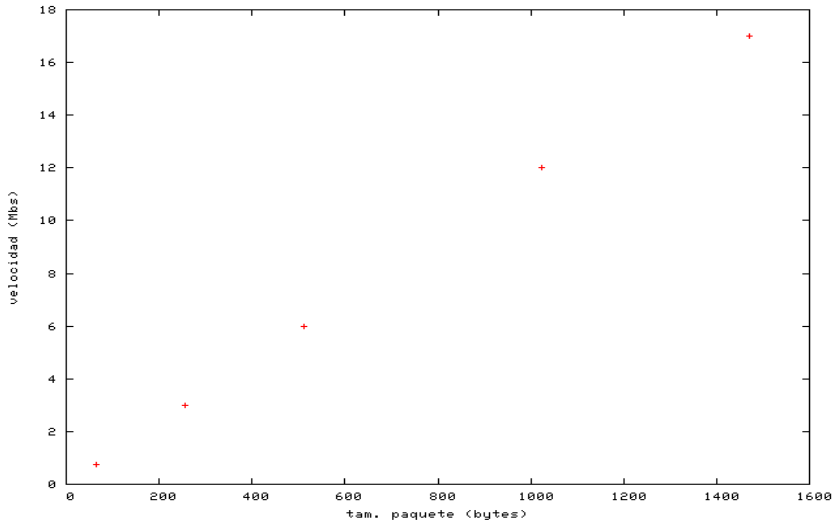


Figura 30 Gráfica velocidad máxima sin perdidas según el tamaño de paquete

La figura 31 muestra la relación entre el tamaño de paquete y el número de paquetes por segundo, en el momento que la aplicación está al límite de procesado:

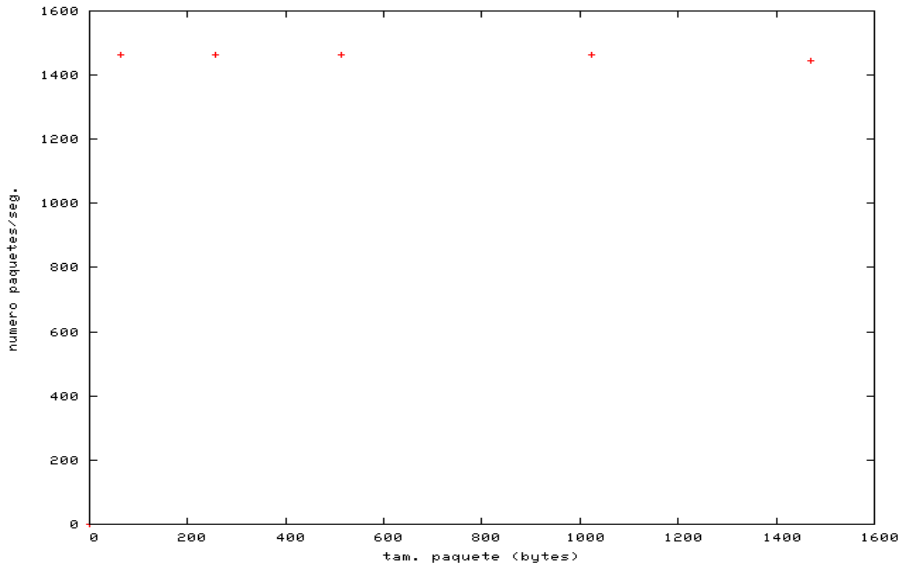


Figura 31 Número de paquetes procesados por segundo según el tamaño de paquete

Se observa que S.S.PYMES es capaz de procesar muchos menos paquetes que Ntop, y todo es debido al límite que MySQL pone en el procesado de paquetes, ya que no es capaz de procesar más de 16.000 transacciones por segundo más o menos y por lo tanto. Además mejoras como el agrupamiento de sentencias no se han podido poner en práctica debido a que la estructura de la implementación no lo permite.

9. CONCLUSIONES

Con el estudio realizado sobre Ntop se puede comprobar que es una herramienta muy eficiente, en cuanto a la pérdida de paquetes (es capaz de procesar paquetes de tamaño medio de 200 bytes a 100mb/seg.),la información que muestra, es muy completo. Sin embargo, debido a la gran información de datos que muestra, la cantidad de gráficas, y todo el procesamiento que conlleva hace que pierda un poco de efectividad.
En principio se pensaba que el uso de BBDD en un sistema de monitorización, en nuestro caso S.S.PYMES, era suficiente, ya que se encontraron algunas optimizaciones que mejoraban los resultados, pero a la hora de implementar, nos topamos con algunos problemas que no han dejado implementarlas, lo que hace que finalmente S.S.PYMES sea capaz de procesar paquetes de tamaño medio de 200 bytes a 3mb/seg. .
Por lo tanto, el uso de BBDD en un sistema de monitorización no es suficiente si este sistema es de gran escala. Es debido a que el procesamiento de MySQL limita el proceso de captura y análisis de paquetes, es decir, que un programa capturaría y procesaría más paquetes si las operaciones de éste con la BD fueran más rápidas. Ya lo comprobamos en la gráfica 29, que lo que hace trabajar la CPU es MySQL porque el consumo del programa de captura y procesamiento de paquetes es despreciable. Pero si el sistema es de pequeña escala se pueden utilizar BBDD para almacenar la información.
Una de las opciones que Ntop no posee son gráficas de bits por segundo por usuario. Por es S.S.PYMES es capaz de mostrar de cada IP su tráfico entrante y saliente de dos formas, tanto por número de paquetes por segundo como por bits por segundo.
Además de esta información que proporciona S.S.PYMES, podría ofrecer mucha más y de diferentes maneras (tablas, gráficas, sectores, valores, etc), ya que toda esa información está almacenada en la BD.

BIBLIOGRAFÍA

-[1] Página de inicio de NTOP www.ntop.org
-[2] Página de inicio de Apache www.apache.org
-[3] Página de inicio de MySql www.mysql.com
-[4] Página de inicio de Php www.php.net
-[5] Información para Iperf http://seguridadyredes.nireblog.com/post/2008/06/18/iperf-midiendo-ancho-de-banda-entre-dos-hosts
-[6] Información para gnuplot http://www.gnuplot.info/
-[7] Programar con la librería Libpcap www.e-ghost.deusto.es/docs/2005/conferencias/pcap.pdf
-[8] Página de inicio de la Organización TCPDUMP www.tcpdump.org/ -> tcpdump y filtros del programa con libpcap
-[9] Uso de IPERF http://bytecoders.homelinux.com/content/medir-la-red-con-iperf.html
-[10] Ayuda sobre información tarjeta de red, ETHTOOL
http://linux.about.com/od/commands/l/blcmdl8_ethtool.htm
-[11] Contiene paquetes rpm para fedora, redhat,etc. http://rpm.pbone.net/
-[12] Comandos en linux http://www.esdebian.org/wiki/lista-comandos-gnulinux-ii
-[13] Más comandos en linux http://www.linuxtotal.com.mx/index.php?cont=info_admon_015
-[14] Comandos linux información del sistema http://www.aprendergratis.com/viendo-la-informacion-del-sistema-en-linux.html
-[15] Funciones de la librería libpcap http://www.winpcap.org/docs/docs_40_2/html/group__wpcapfunc.html
-[16] Manual de Mysql http://www.webestilo.com/mysql/
-[17] Manual de C http://www.fismat.umich.mx/mn1/manual/
-[18] Tutorial Php y Mysql
http://www.wikilearning.com/tutorial/tutorial_de_php_y_mysql_funciones_de_acceso_a_ficheros/9869-13
-[19] Socket Unix y socket TCP/IP http://dev.mysql.com/doc/refman/5.0/es/can-not-connect-to-server.html
-[20] Página de inicio de Gnuplot http://www.gnuplot.info/